



Rapport du projet UML et Java EE

Génie informatique

Sujet du projet :
Location de voitures

Réalisé par :
Marwa BOUMOUR
El Mimouni Yosra

Résumé

En résumé, ce projet vise à concevoir et développer **une application web côté serveur** innovante dédiée à la gestion d'une plateforme de location de voitures. L'objectif principal est de simplifier et d'optimiser les processus de consultation, réservation, location et gestion des véhicules. Cette application permet aux utilisateurs, qu'ils soient clients ou administrateurs, de bénéficier d'une expérience fluide et intuitive pour interagir avec le système.

Les fonctionnalités principales incluent la gestion des utilisateurs (authentification, gestion de profil), la gestion des véhicules (ajout, mise à jour, suppression et consultation des véhicules disponibles), ainsi que la réservation en ligne, intégrant des filtres avancés comme les dates, types de véhicules et d'autres critères. De plus, le projet intègre un module de paiement et de facturation, garantissant le calcul précis des coûts et la génération automatique des factures.

Cette plateforme est conçue selon des principes méthodologiques modernes tels que Scrum pour la gestion du projet et UML pour la modélisation. Elle met également en œuvre des technologies robustes comme Java EE, Spring Boot et Thymeleaf pour le backend, avec une approche centrée sur l'efficacité, la sécurité et la convivialité.

Contents

Résumé	1
1 Présentation de projet	3
1.1 Objectif du Projet	3
1.2 Définition du Projet	3
2 Etude conceptuelle	4
2.1 Introduction	4
2.2 La méthode Agile	5
2.2.1 Aperçu des méthodologies Agile	5
2.3 Conception UML	6
2.3.1 définition	6
2.3.2 Diagramme de Cas d'Utilisation détaillés	8
2.3.3 Diagramme de Classe	9
2.3.4 Diagramme d'Activité : Processus de Réservation et Validation	10
2.3.5 Diagramme de Séquence : Processus d'Authentification avec Spring Security	11
2.3.6 Conclusion	11
3 Technologies Utilisées	11
3.1 Environnement de Développement Intégré (IDE)	11
3.1.1 Définition :	11
3.1.2 IntelliJ IDEA	12
3.2 Serveur d'Applications	12
3.2.1 Apache Tomcat	12
3.3 Langages de Programmation et Bibliothèques	13
3.3.1 Front End	13
3.3.2 Back End	16
3.4 Bases de Données	17
4 Réalisation :	20
4.1 Partie Client	20
4.1.1 Page d'accueil	20
4.1.2 Demande de reservation	21
4.2 Partie Admin	28

1 Présentation de projet

1.1 Objectif du Projet

L'objectif principal du projet est de concevoir et de développer une application web de location de voitures. Cette application permet aux utilisateurs de consulter, réserver, louer, et gérer des véhicules. Elle vise à simplifier les opérations de gestion pour une entreprise de location, notamment en ce qui concerne les voitures, les clients, les réservations, et les paiements.

L'application repose sur deux axes principaux :

Une interface intuitive et facile d'utilisation pour les clients souhaitant louer des véhicules.

Un système performant et centralisé pour les administrateurs permettant la gestion des réservations, des paiements, et des rapports d'utilisation.

1.2 Définition du Projet

L'application est construite en utilisant Java EE pour la gestion du backend, complété par des outils comme Spring Boot pour optimiser le développement. Le frontend repose sur Thymeleaf, une bibliothèque intuitive pour le rendu des pages. Le système de gestion de la base de données est assuré par MySQL, avec une configuration Docker pour faciliter le déploiement.

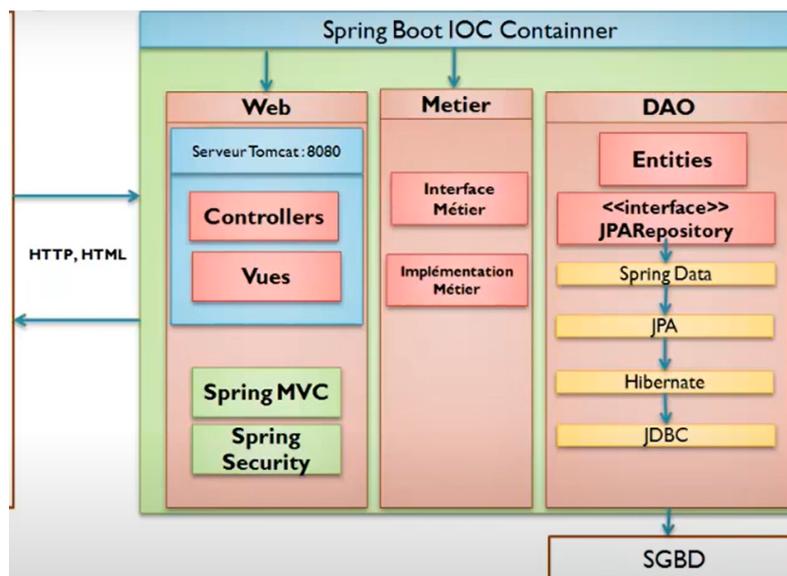


Figure 1: schéma simplifiant l'architecture de notre projet

2 Etude conceptuelle

2.1 Introduction

L'axe de l'étude conceptuelle constitue la base fondamentale sur laquelle repose la conception de l'application de gestion des transports au sein de la SOTREG. Cette phase préliminaire, essentielle à la réussite du projet, consiste à approfondir les principes conceptuels qui orienteront le développement de l'application. **Grâce aux visites effectuées dans plusieurs entreprises de gestion de véhicules à Khouribga**, nous avons pu mieux appréhender les défis spécifiques du secteur et les meilleures pratiques à adopter. Ces échanges ont permis d'identifier les contraintes et opportunités liées à la gestion du transport du personnel et de définir les principales lignes directrices de notre projet.

L'objectif de cette étude conceptuelle est de décomposer et d'analyser les éléments essentiels du projet, en mettant l'accent sur les besoins des utilisateurs, les caractéristiques du système de transport et les contraintes techniques. Cette analyse permet également de clarifier les problématiques à résoudre, comme l'optimisation des trajets et la gestion de la disponibilité des véhicules, afin d'offrir une solution efficace et adaptée aux attentes de la SOTREG.

L'étude conceptuelle est donc une étape clé pour s'assurer que l'application proposée réponde aux besoins réels des utilisateurs tout en respectant les objectifs stratégiques du projet.



Figure 2: Parmi les entreprises qu'on a visité

2.2 La méthode Agile

lité, la collaboration et l'adaptation continue aux changements. Elle est conçue pour répondre aux défis des projets complexes et évolutifs, en favorisant des interactions régulières avec les parties prenantes et en encourageant une livraison incrémentielle et itérative des résultats. Les méthodes agiles sont particulièrement adaptées aux environnements où les exigences évoluent rapidement et où une approche plus traditionnelle pourrait ne pas être aussi efficace. La méthodologie Agile s'oppose généralement à la méthodologie traditionnelle "Waterfall", ou dite "de cycle en V". Plus souple et plus flexible, elle place les besoins du client au centre des priorités du projet. Lors de sa création, cette approche devait servir à la gestion des projets de développement web et informatique. Aujourd'hui, la méthode Agile est de plus en plus répandue. Ce succès s'explique notamment par son adaptabilité à de nombreux types de projets, tous secteurs confondus

2.2.1 Aperçu des méthodologies Agile

Les fondements de la méthodologie Agile ont été développés par 17 personnes en 2001 sous forme écrite. Leur Manifeste Agile pour le développement de logiciels mettait en avant une approche novatrice de la création de valeur ajoutée et de la collaboration avec les clients. Les quatre principales valeurs de la méthodologie Agile sont les suivantes :

- Les individus et leurs interactions, de préférence aux processus et aux outils.
- Des solutions opérationnelles, de préférence à une documentation exhaustive.
- La collaboration avec les clients, de préférence aux négociations contractuelles.
- La réponse au changement, de préférence au respect d'un plan.

Pour mettre en œuvre la méthode Agile, différentes approches et cadres de travail sont utilisés, tels que Scrum, Kanban, Extreme Programming (XP) et bien d'autres. Voici un aperçu rapide de **Scrum**, l'une des méthodes agiles les plus populaires :

Scrum

Scrum divise un projet en cycles de développement appelés "itérations" (ou "sprints"). Chaque itération dure généralement de 1 à 4 semaines. Pendant chaque sprint, l'équipe se concentre sur la livraison d'un ensemble de fonctionnalités spécifiques. Les principales caractéristiques de Scrum incluent :

- **Rôles** : Scrum Master (responsable du processus), Product Owner (représentant du client) et Équipe de développement.

- **Artéfacts** : Backlog de produit (liste des fonctionnalités à développer) et Backlog de sprint (liste des fonctionnalités sélectionnées pour le sprint en cours).
- **Cérémonies** : Réunion de planification de sprint, réunion quotidienne de standup, revue de sprint et rétrospective de sprint.

L'équipe travaille de manière autonome et auto-organisée, en se concentrant sur la collaboration et l'itération continue pour fournir des résultats de haute qualité.

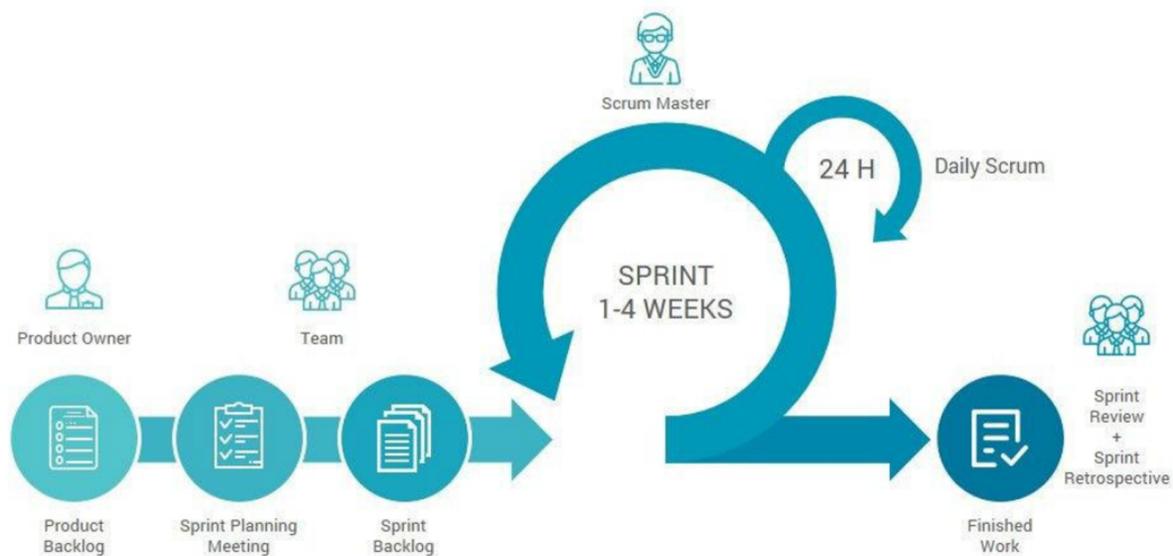


Figure 3: Méthode Scrum

En somme, la méthode agile vise à fournir de la valeur de manière itérative, à favoriser la collaboration et à s'adapter aux changements, ce qui en fait une approche précieuse pour la gestion de projets complexes et évolutifs.

2.3 Conception UML

2.3.1 définition

L'UML (Unified Modeling Language) est un langage de modélisation standard utilisé en ingénierie logicielle pour représenter, concevoir et documenter visuellement les systèmes logiciels. Il fournit un vocabulaire commun et un ensemble de notations graphiques pour modéliser différents aspects de la structure, du comportement et des interactions d'un système. Les diagrammes UML sont largement utilisés tout au long du cycle de vie du développement logiciel, de la collecte et de l'analyse des besoins à la conception, la mise en œuvre et les tests.

L'UML propose une large gamme de types de diagrammes, chacun servant un objectif spécifique dans la capture de différentes facettes d'un système. Voici quelques-uns des diagrammes UML les plus couramment utilisés :

- **Diagramme de Classes** : Représente la structure statique d'un système en montrant les classes, leurs attributs, leurs méthodes, leurs relations et leurs hiérarchies d'héritage.
- **Diagramme de Cas d'Utilisation** : Décrit la fonctionnalité ou le comportement d'un système du point de vue de l'utilisateur, en capturant les interactions entre les utilisateurs (acteurs) et le système à travers des cas d'utilisation.
- **Diagramme d'Activité** : Modélise le flux d'activités ou de processus au sein d'un système, montrant la séquence d'actions, les décisions et les activités concurrentes impliquées dans un processus particulier.
- **Diagramme de Séquence** : Illustre les interactions entre les objets ou les composants d'un système sur une période de temps spécifique, montrant l'ordre des échanges de messages et les lignes de vie des objets participants.
- **Diagramme d'État-Transition** : Représente les différents états et transitions d'un objet ou d'un système, montrant comment il répond aux événements et évolue avec le temps.
- **Diagramme de Composants** : Dépeint les composants physiques et logiques d'un système ainsi que leurs dépendances, mettant en évidence l'organisation et les relations entre différentes parties du système.
- **Diagramme de Déploiement** : Montre le déploiement physique des composants logiciels sur les nœuds matériels, illustrant comment le système est réparti sur différentes machines ou appareils.

float

2.3.2 Diagramme de Cas d'Utilisation détaillés

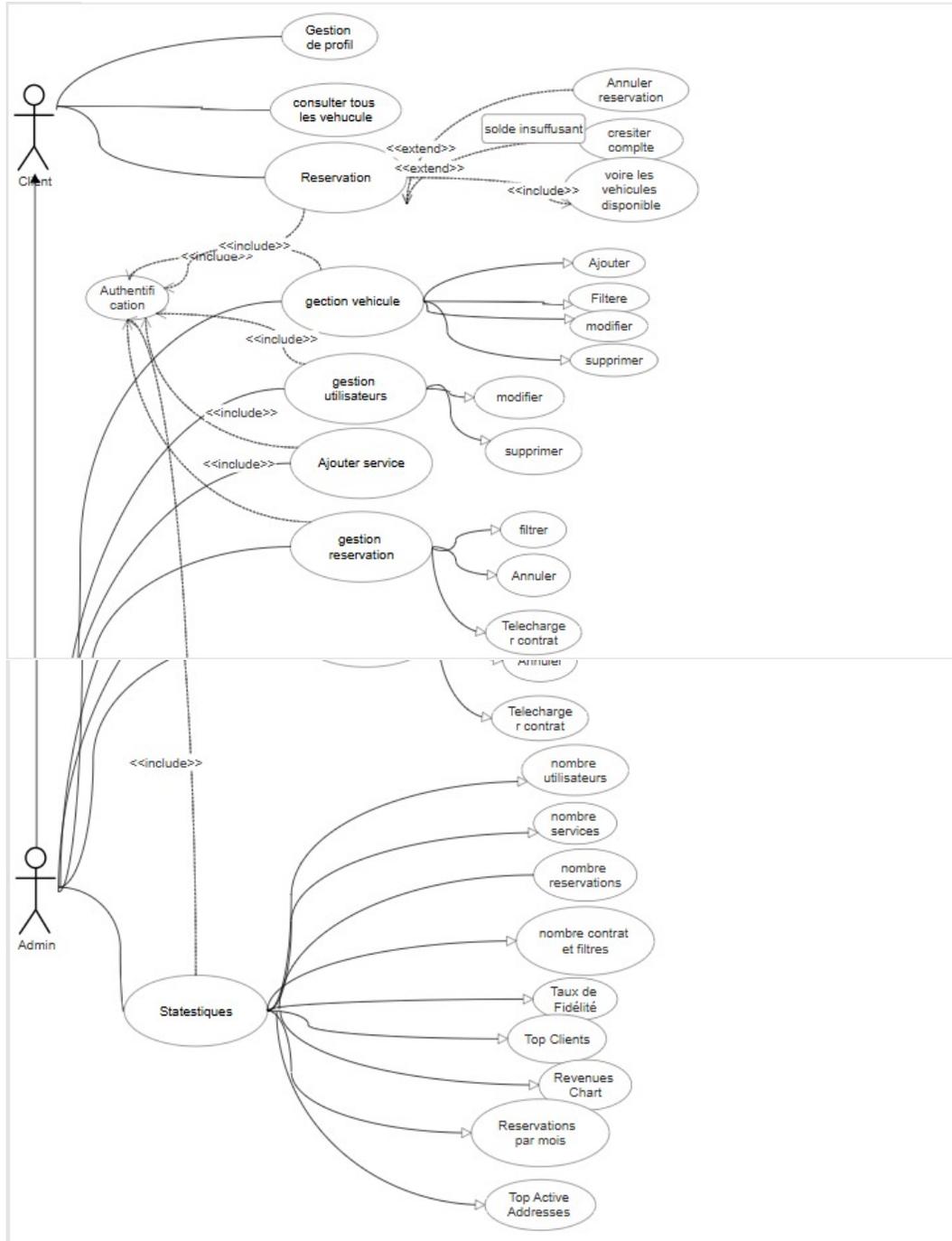


Figure 4: diagramme de cas d'utilisation détaillé.

2.3.3 Diagramme de Classe

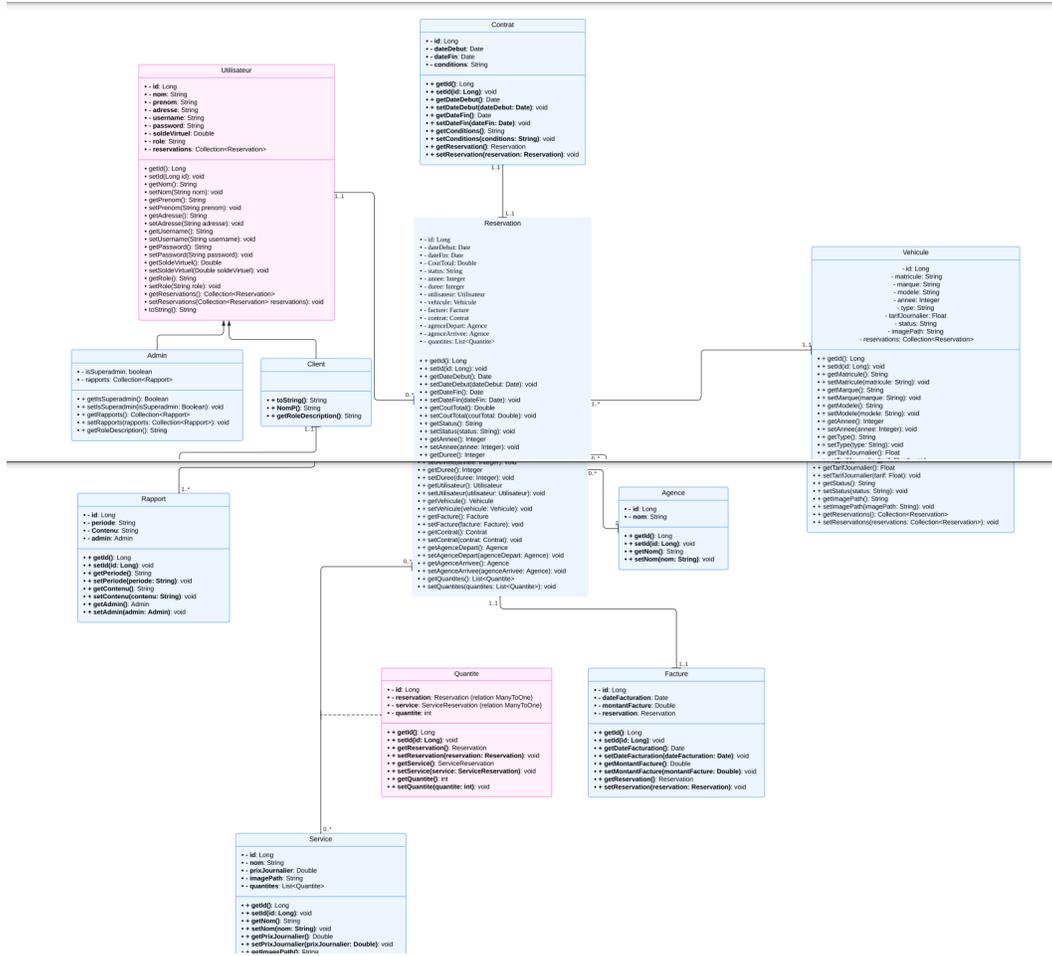


Figure 5: Diagramme de Classe divisé en deux parties

2.3.4 Diagramme d'Activité : Processus de Réservation et Validation

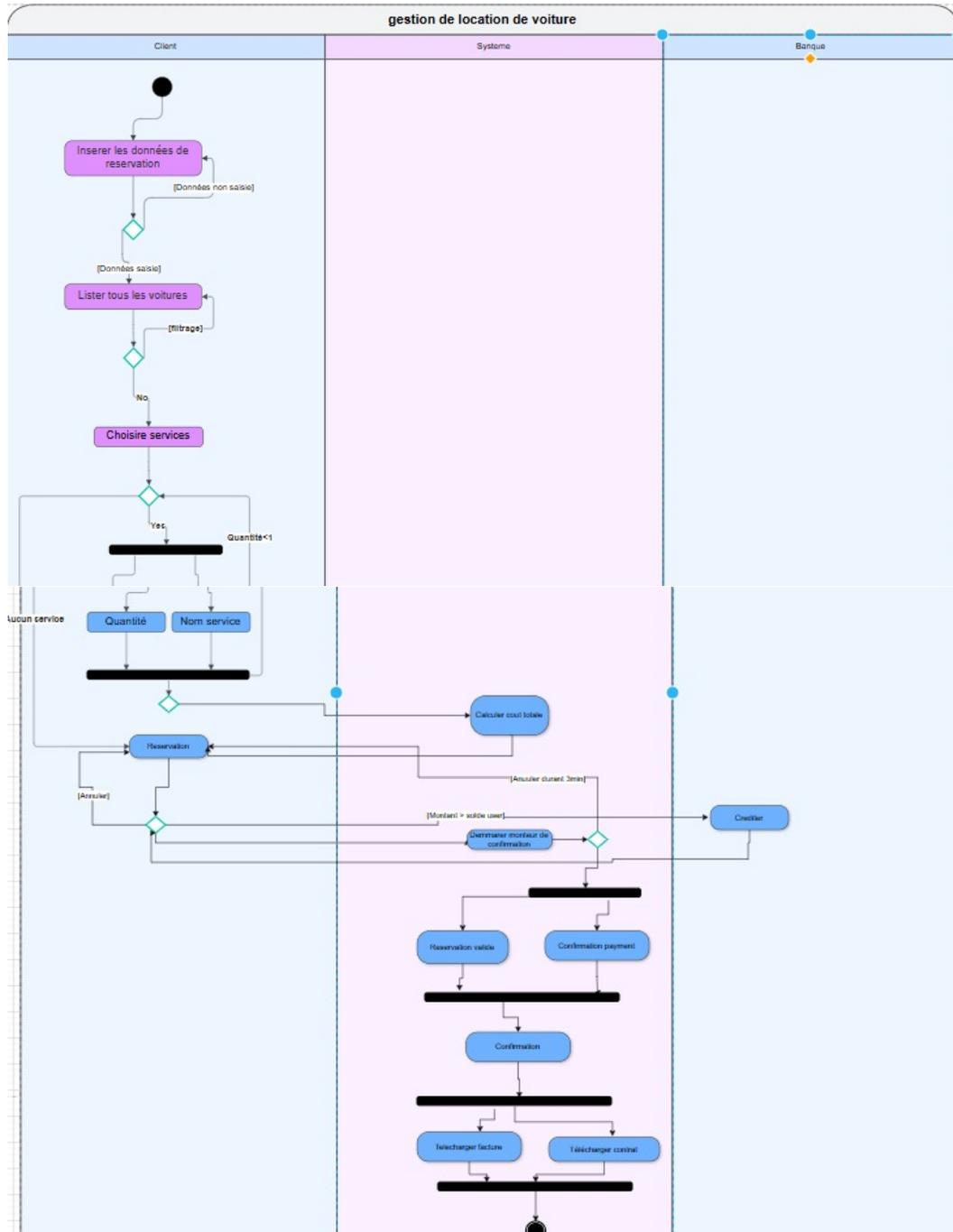


Figure 6: Processus de Réservation et Validation

2.3.5 Diagramme de Séquence : Processus d'Authentification avec Spring Security

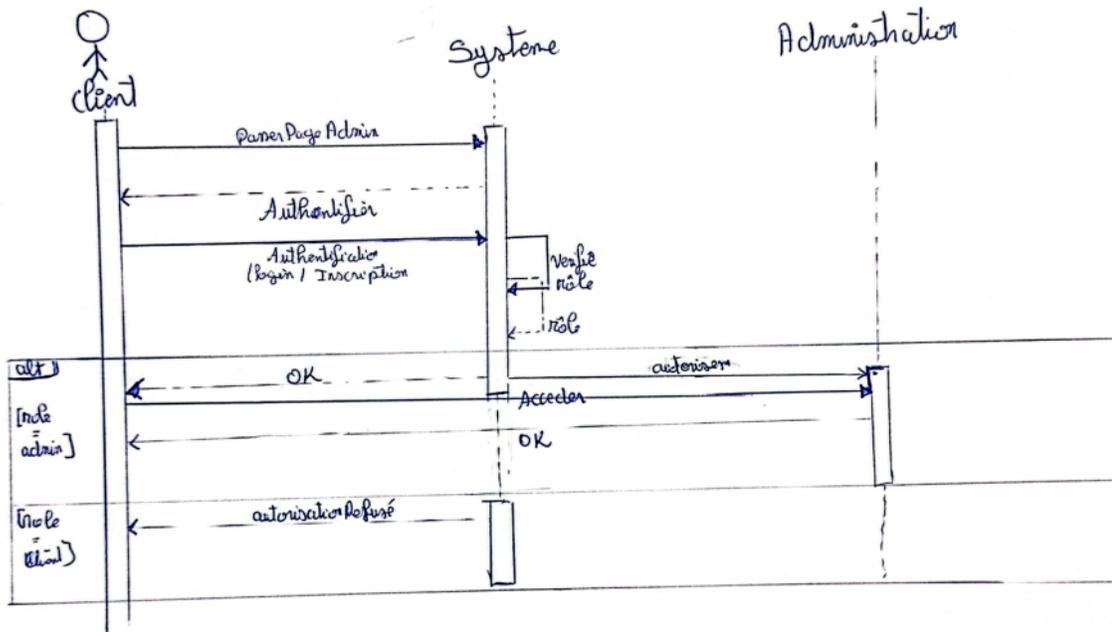


Figure 7: Logo Spring Boot

2.3.6 Conclusion

En fait, la conception enrichit la compréhension de la méthodologie de développement. Cela nous permettra, dans le chapitre suivant, d'aborder la phase de développement et la simulation de notre application.

3 Technologies Utilisées

3.1 Environnement de Développement Intégré (IDE)

3.1.1 Définition :

Un Environnement de Développement Intégré (IDE) est une suite logicielle qui regroupe les outils de base nécessaires à l'écriture et au test de logiciels. Il s'agit d'une application logicielle qui fournit des fonctionnalités complètes aux programmeurs pour le développement de logiciels. Un IDE comprend généralement un éditeur de code source, des outils d'automatisation de la construction et un débogueur, accessibles via une interface utilisateur graphique (GUI) unique.

3.1.2 IntelliJ IDEA

IntelliJ IDEA est un Environnement de Développement Intégré (IDE) pour Java, Kotlin, Groovy et d'autres langages basés sur la machine virtuelle Java (JVM). Il est développé et maintenu par JetBrains et est disponible en tant qu'édition communautaire et édition ultime. IntelliJ IDEA est l'un des IDE les plus puissants et populaires pour Java. C'est un IDE riche en fonctionnalités qui permet un développement rapide et contribue à améliorer la qualité du code.

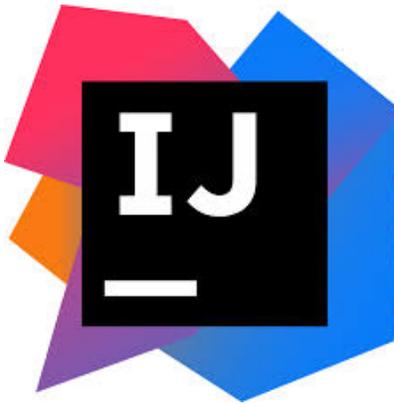


Figure 8: Icône IntelliJ IDEA

3.2 Serveur d'Applications

3.2.1 Apache Tomcat

Apache Tomcat est un serveur d'applications open-source conçu pour exécuter des applications web Java. Il implémente les spécifications des technologies Java Servlet, JavaServer Pages (JSP) et WebSocket, fournies par Java EE (Jakarta EE).

Dans notre projet de gestion de location de voitures, Tomcat joue un rôle central en tant que conteneur d'applications pour le déploiement et l'exécution des services web. Grâce à sa simplicité et à sa flexibilité, il est particulièrement adapté pour les projets nécessitant un serveur léger mais robuste.

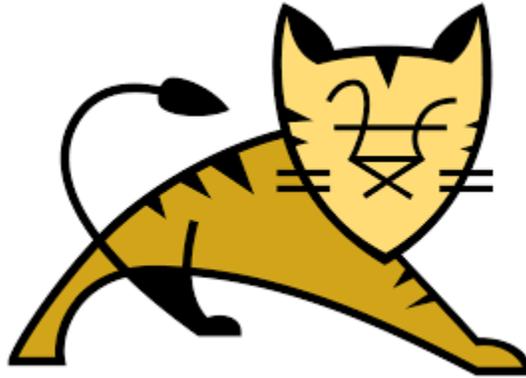


Figure 9: Icône du serveur Apache Tomcat

3.3 Langages de Programmation et Bibliothèques

3.3.1 Front End

Pour le développement front end de notre application, nous avons utilisé les technologies suivantes :

HTML : HTML, ou HyperText Markup Language (langage de balisage hypertexte), est le langage fondamental utilisé pour créer et structurer le contenu des pages web. En tant que langage de balisage, il utilise des balises pour définir la structure et le format du contenu sur une page web. HTML permet de décrire la structure logique d'une page, c'est-à-dire comment le contenu est organisé, quels éléments sont des titres, des paragraphes, des liens, des images, des listes, etc.



Figure 10: LOGO HTML

CSS : CSS, ou Cascading Style Sheets (feuilles de style en cascade), est un langage de style utilisé pour décrire la présentation d'un document écrit en HTML ou en XML. CSS permet

de contrôler la mise en page, les couleurs, les polices, les espacements et de nombreux autres aspects de l'apparence des pages web. Il permet également de séparer le contenu de la présentation, facilitant ainsi la maintenance et la flexibilité du design web.



Figure 11: LOGO CSS

Bootstrap : Bootstrap est un framework front end open source qui facilite le développement rapide et responsive des sites web et des applications web. Il fournit un ensemble de composants HTML, CSS et JavaScript prêts à l'emploi, tels que des grilles, des boutons, des formulaires, des modales et bien plus encore. Bootstrap aide à créer des designs modernes et cohérents avec une compatibilité multi-navigateurs et multi-appareils.



Figure 12: LOGO Bootstrap

Thymeleaf : Thymeleaf est un moteur de templates Java pour le développement web, qui permet de générer des pages dynamiques côté serveur. Dans notre application, Thymeleaf a été utilisé pour intégrer dynamiquement des données backend dans les vues HTML. Grâce à son intégration native avec Spring Boot et Java EE, il offre une syntaxe simple pour afficher des données, gérer des boucles et des conditions, et inclure des fragments.



Figure 13: Logo Thymeleaf

3.3.2 Back End

Pour le développement back end de notre application, nous avons utilisé les technologies suivantes :

Spring Boot : Spring Boot est un framework Java open source utilisé pour créer des applications basées sur Spring de manière rapide et avec une configuration minimale. Il permet de simplifier le processus de développement en offrant des fonctionnalités préconfigurées, ce qui accélère le démarrage des projets. Spring Boot est particulièrement adapté pour la création d'APIs REST et d'applications web avec une architecture de microservices.



Figure 14: Logo Spring Boot

Spring Security : Spring Security est un framework puissant et flexible pour sécuriser les applications Java. Dans notre application, nous avons intégré Spring Security pour gérer l'authentification et l'autorisation des utilisateurs. Spring Security offre plusieurs mécanismes pour la gestion des utilisateurs, y compris la sécurité basée sur les rôles, l'intégration avec des bases de données pour stocker les informations d'authentification, et la protection contre les attaques courantes telles que les attaques par injection SQL et CSRF.

Nous avons opté pour une gestion **stateful** des sessions pour garder une trace de l'utilisateur pendant sa session, ce qui permet de maintenir une connexion persistante au serveur pendant l'interaction avec l'application.

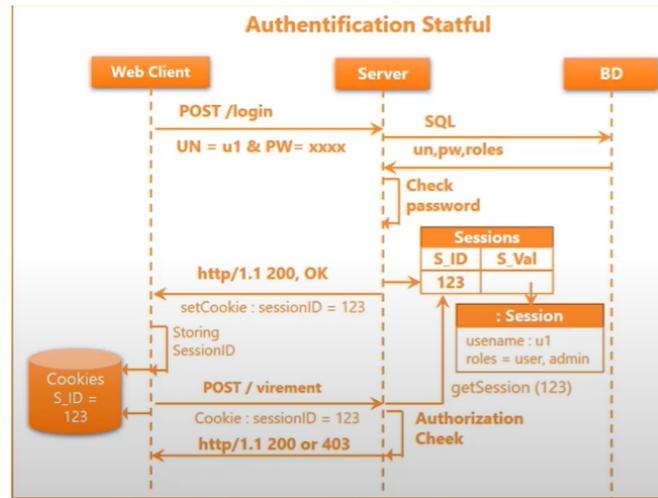


Figure 15: Logo Spring Security Stateful

Gestion de sessions (Stateful et Stateless) :

- **Stateful** : Dans un environnement stateful, chaque requête utilisateur est liée à une session spécifique sur le serveur, ce qui permet de suivre l'état de la session (comme l'authentification). Le serveur garde une trace des informations de session entre les requêtes, offrant une expérience utilisateur continue.
- **Stateless** : À l'inverse, une approche stateless ne garde aucune information entre les requêtes. Chaque requête est indépendante, ce qui peut être plus scalable pour les applications de grande envergure, mais nécessite souvent un système de tokens (comme JWT) pour maintenir l'état de l'authentification.

Dans notre application, nous avons opté pour une gestion **stateful** des sessions utilisateur, utilisant des cookies de session pour maintenir l'état de l'utilisateur pendant ses interactions avec l'application.

3.4 Bases de Données

La gestion des bases de données est cruciale dans les applications, car elle permet de gérer efficacement les données et d'effectuer diverses tâches de manière fluide. Un système de gestion de bases de données (SGBD) permet de stocker, organiser et surveiller les informations, qu'elles soient volumineuses ou réduites, à travers une application logicielle unique. Il offre aux organisations une vue consolidée et claire sur la gestion des données, évitant ainsi les duplications inutiles et facilitant la mise en place de politiques de sécurité et de confidentialité des données pour réduire les risques de violations.

Avec l'augmentation des volumes de données, la gestion des bases de données devient indispensable pour maintenir la performance des applications, tout en minimisant les impacts sur la disponibilité, la conformité et la sécurité. Les avantages d'un SGBD sont significatifs, notamment en matière de partage sécurisé des données, d'intégration efficace, de cohérence et de fiabilité des informations, ainsi que de respect des normes de confidentialité. Alors que la quantité de données générées par les individus et les machines continue de croître, l'importance des systèmes de gestion de bases de données devient de plus en plus évidente.

MySQL :

MySQL est un système de gestion de bases de données relationnelles open source (SGBDR) qui repose sur le langage de requête structuré (SQL). Il est principalement utilisé pour administrer les données au sein de bases de données et trouve une application fréquente dans les environnements de développement web et la publication en ligne. MySQL constitue un élément clé de la pile open source LAMP, qui se compose de Linux en tant que système d'exploitation, Apache comme serveur web, MySQL pour la gestion des bases de données relationnelles, et PHP comme langage de script orienté objet.

Fonctionnant selon un modèle client-serveur, le serveur MySQL est responsable de la gestion des instructions et des commandes relatives aux bases de données. MySQL est largement adopté dans de nombreuses applications web reposant sur des bases de données, telles que Drupal, Joomla, phpBB et WordPress. De plus, il est employé par de nombreux sites web populaires, dont Facebook, Flickr, MediaWiki, Twitter et YouTube, illustrant ainsi son importance et sa polyvalence dans le monde numérique.



Figure 16: LOGO Mysql

Spring Data JPA :

Spring Data JPA est un module de Spring qui simplifie l'interaction avec la base de données en utilisant JPA (Java Persistence API). Il permet de définir des interfaces Repository pour interagir avec les entités de la base de données sans avoir besoin de coder des requêtes SQL manuelles. Nous avons utilisé Spring Data JPA pour gérer les entités de notre application telles que **Utilisateur**, **Réservation**, et **Facture**, ce qui nous a permis de faciliter la persistance des données et d'améliorer la maintenabilité de l'application.

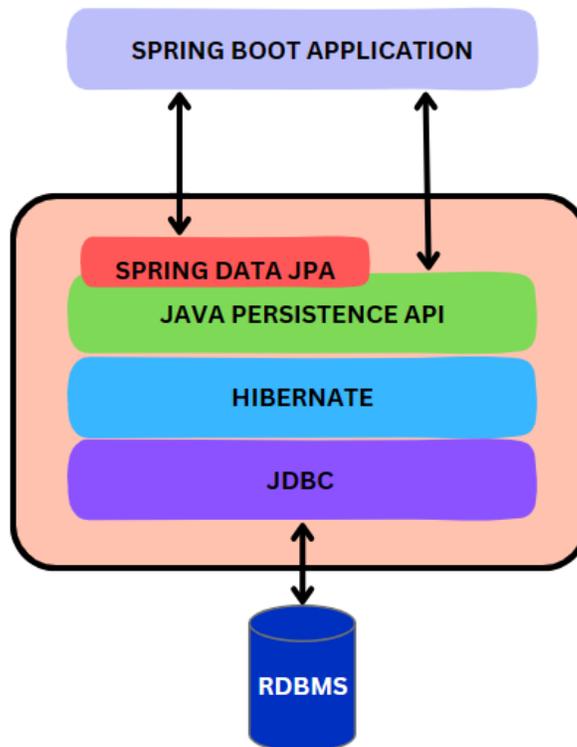


Figure 17: Logo Spring Data JPA

4 Réalisation :

4.1 Partie Client

4.1.1 Page d'accueil



Figure 18: page accueil

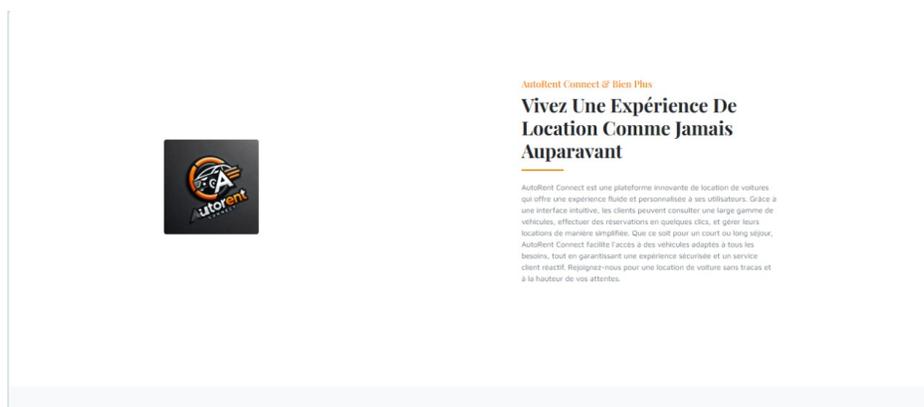


Figure 19: Après avoir cliqué sur le menu "Infos sur notre startup"

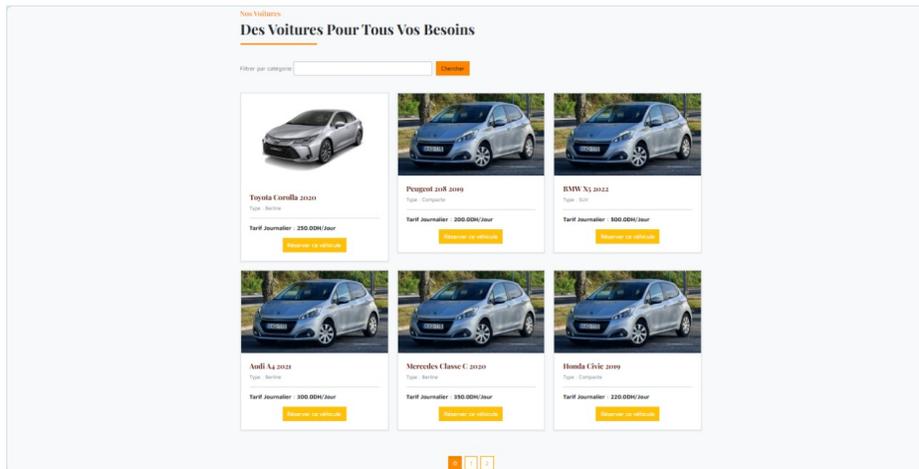


Figure 20: Après avoir cliqué sur le menu "Voir tous les véhicules"

4.1.2 Demande de reservation

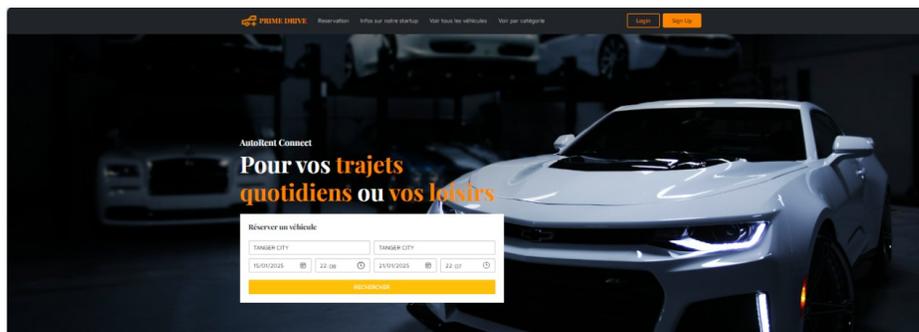


Figure 21

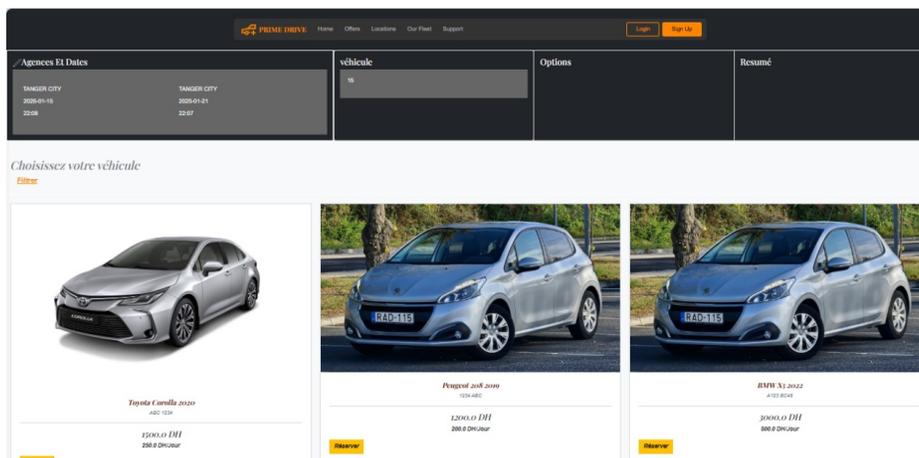


Figure 22

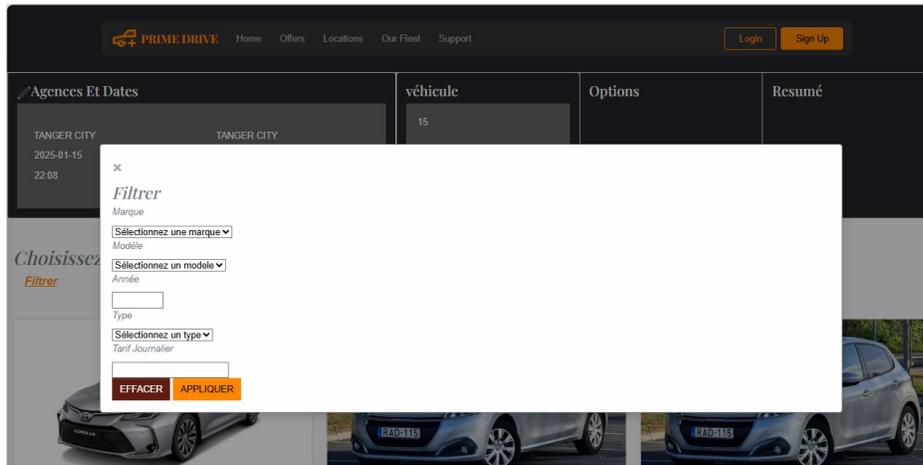


Figure 23

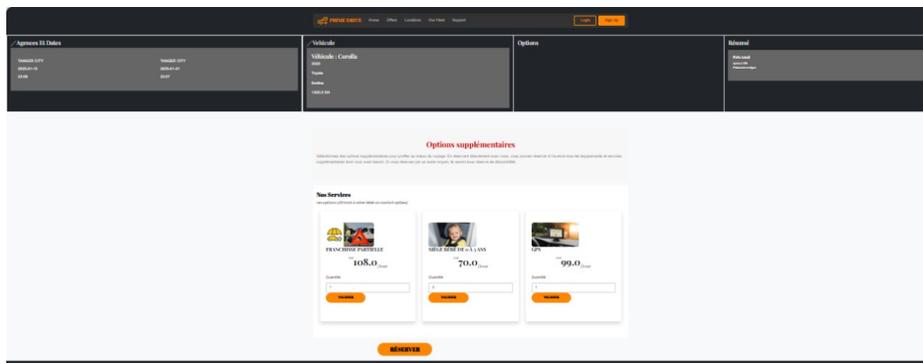


Figure 24

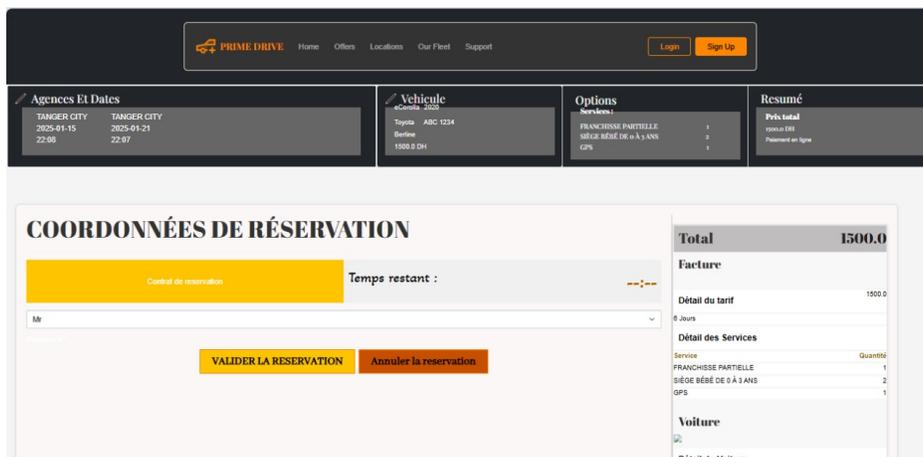


Figure 25

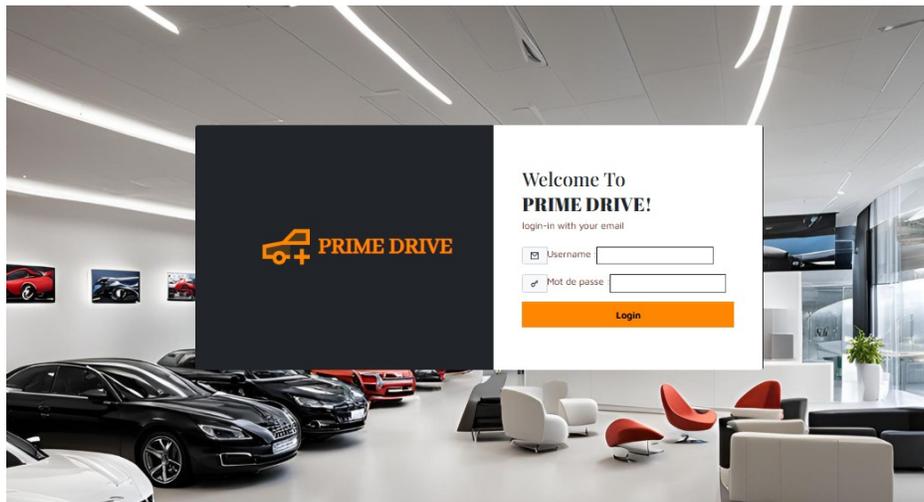


Figure 26

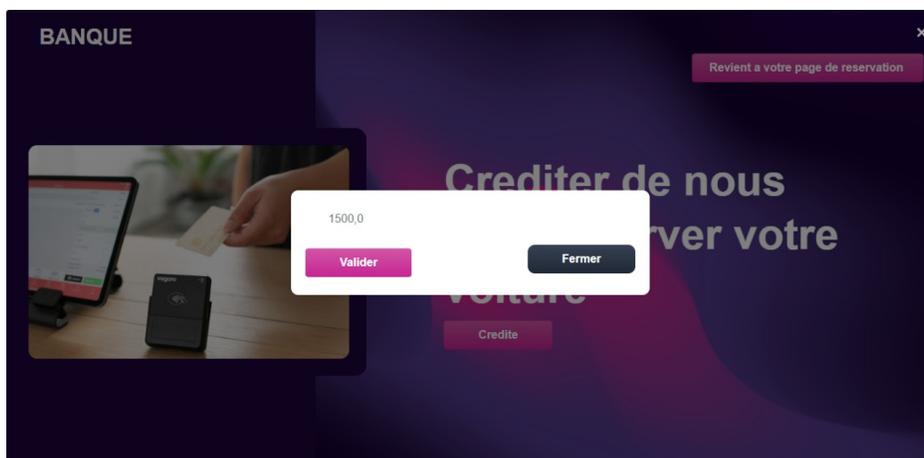


Figure 27

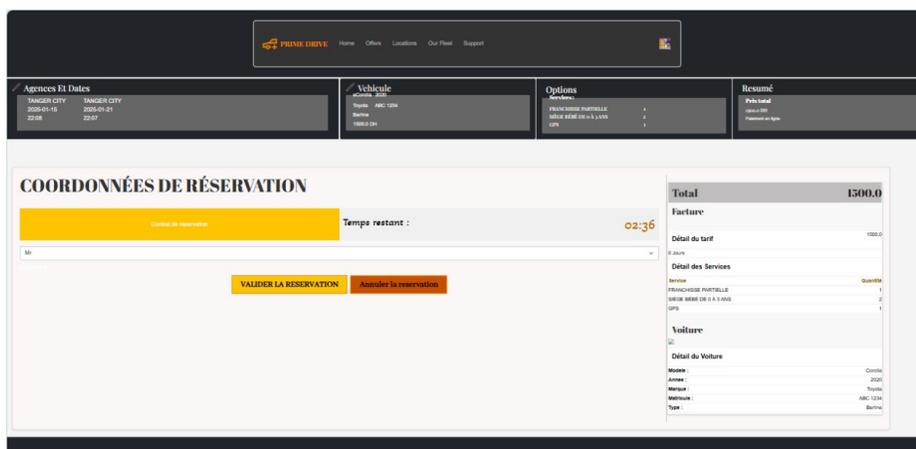


Figure 28

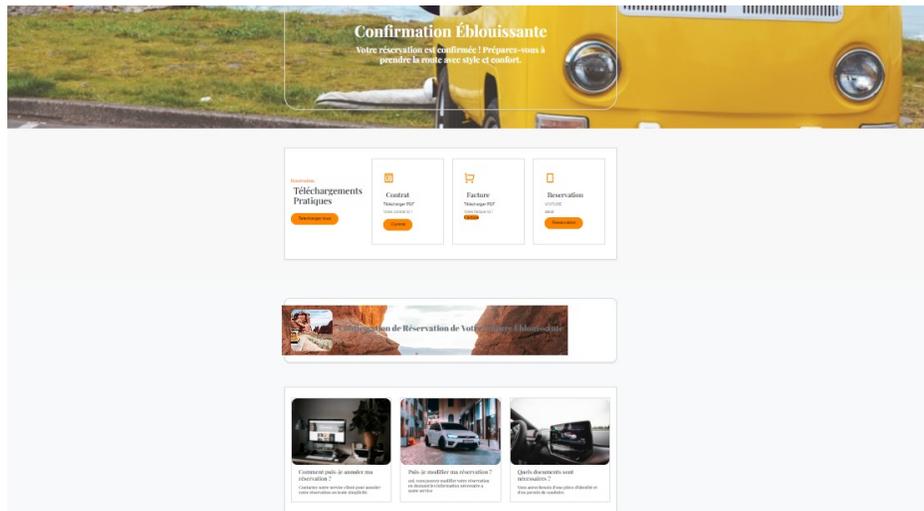


Figure 29

Facture

Détails de la transaction

Détail du tarif:

Durée: 6

Tarif: 1500.0

Services:

Service: FRANCHISSE PARTIELLE - Quantité: 1

Service: SIÈGE BÉBÉ DE 0 À 3 ANS - Quantité: 2

Service: GPS - Quantité: 1

Détails de la voiture:



Modèle: Corolla

Année: 2020

Marque: Toyota

Figure 30

Contrat de Réservation de Voiture

Réservez avec élégance et confort

Modèle: Corolla

Marque: Toyota

Année: 2020

Matricule: ABC 1234

Prix Total: 1500.0 MAD

Service: FRANCHISSE PARTIELLE

Quantité: 1

Service: SIÈGE BÉBÉ DE 0 À 3 ANS

Quantité: 2

Service: GPS

Quantité: 1

Date de début: 2025-01-15

Date d'arrivée: 2025-01-21

Heure d'arrivée: 22:07

Heure de départ: 22:08

Agence de départ: TANGER CITY

Agence d'arrivée: TANGER CITY

Informations de l'utilisateur

Nom de l'utilisateur: Mr mohammed berrada

Adresse: casablanca

email: mohammedberrada@gmail.com

Figure 31

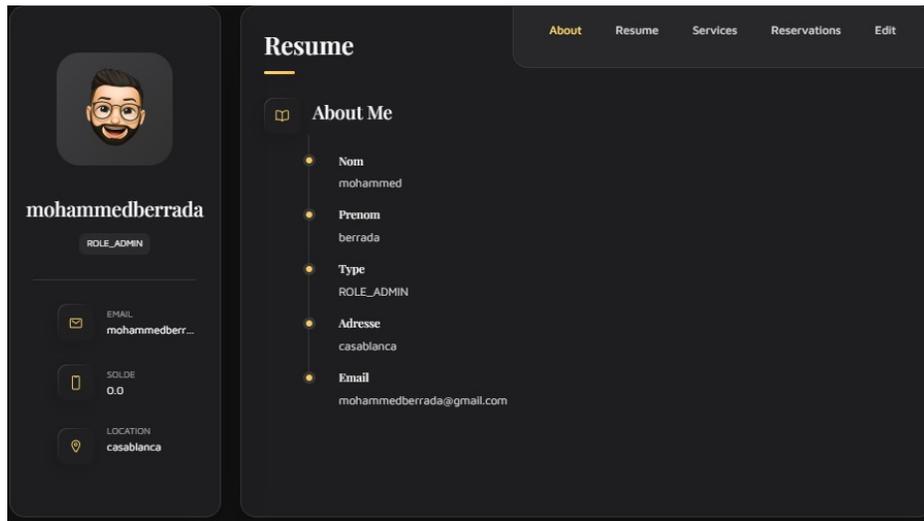


Figure 32

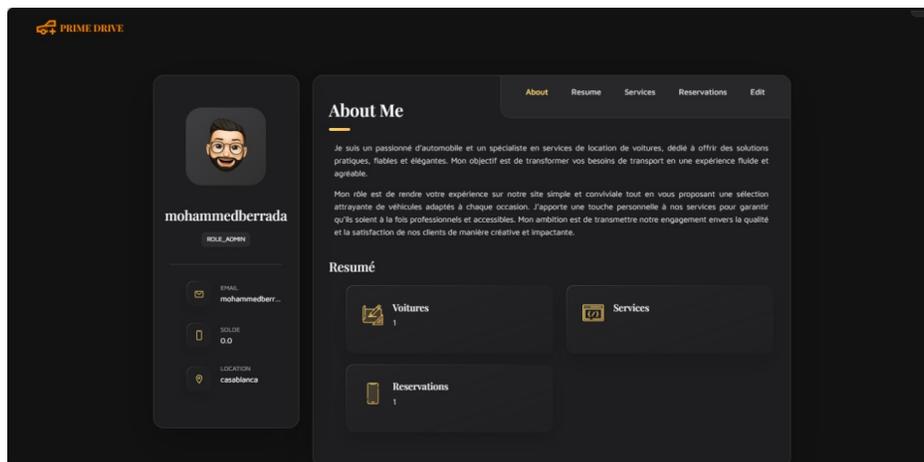


Figure 33

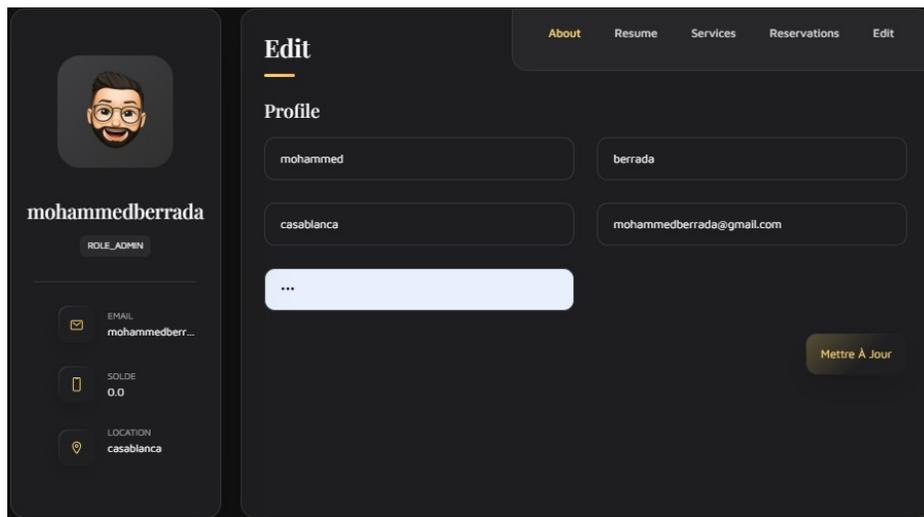


Figure 34

4.2 Partie Admin

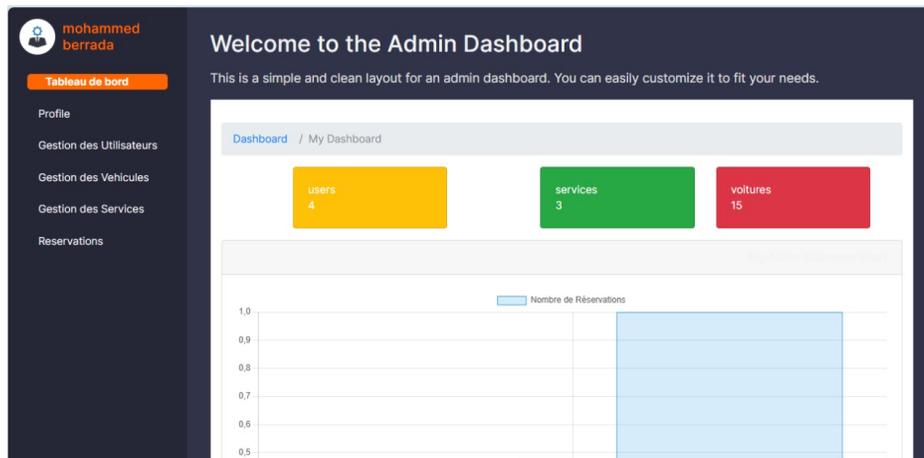


Figure 35



Figure 36

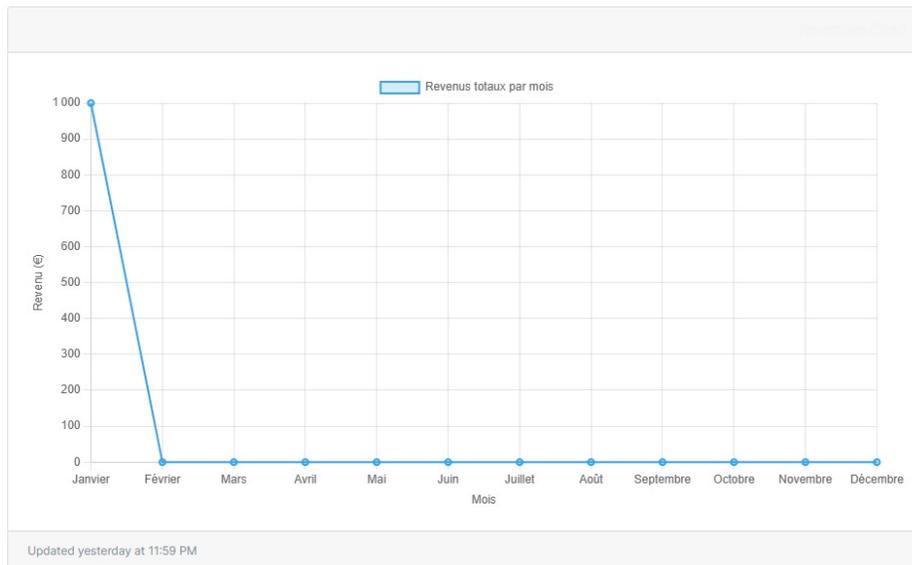


Figure 37

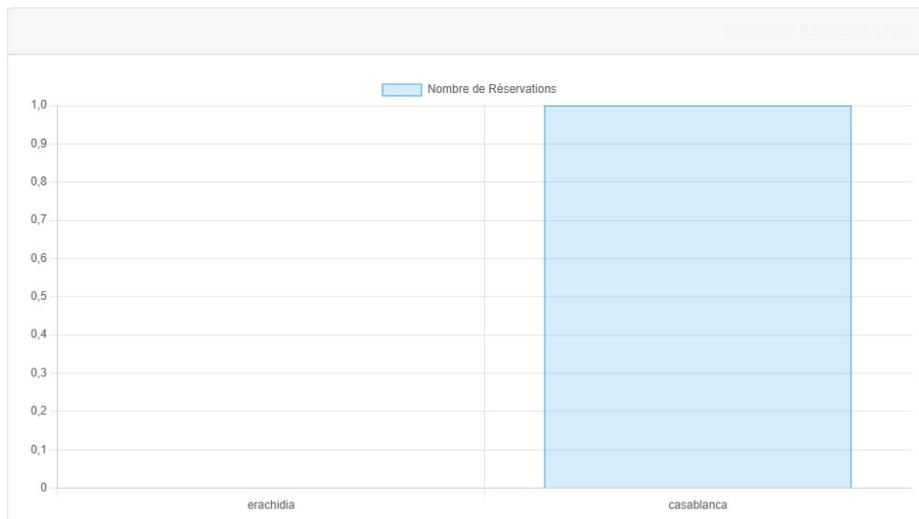


Figure 38

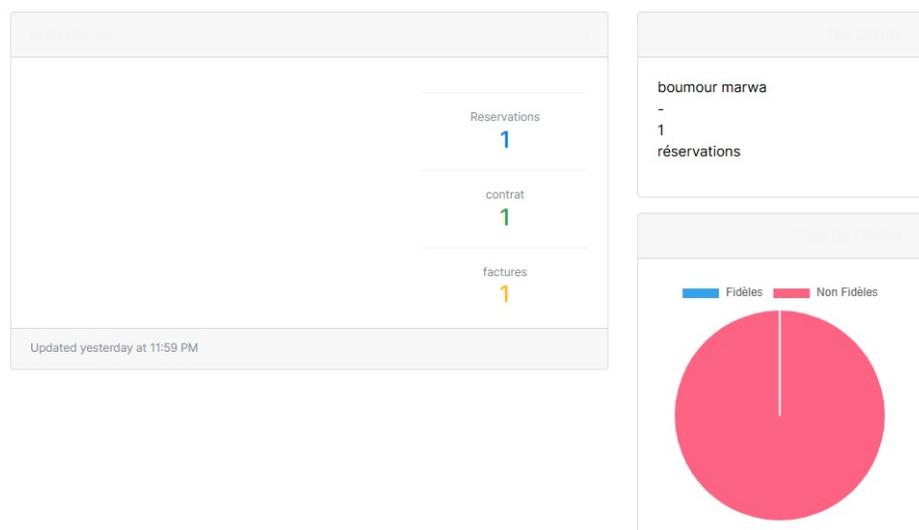
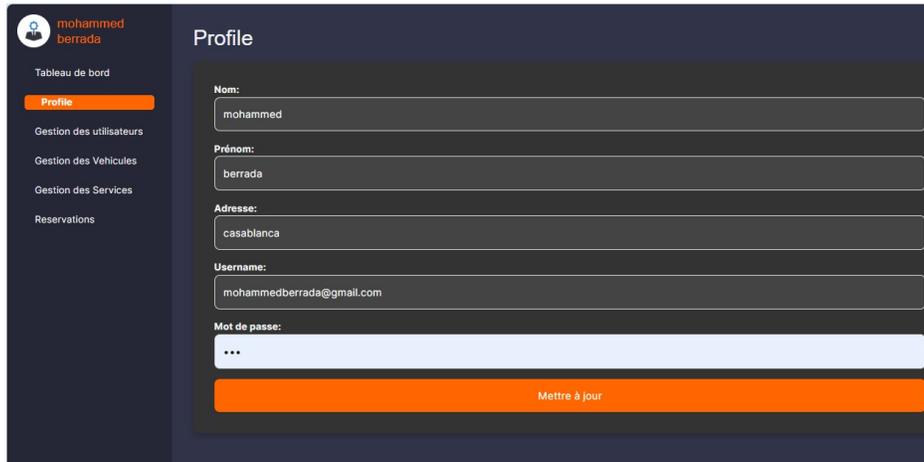


Figure 39



Profile

Nom: mohammed

Prénom: berrada

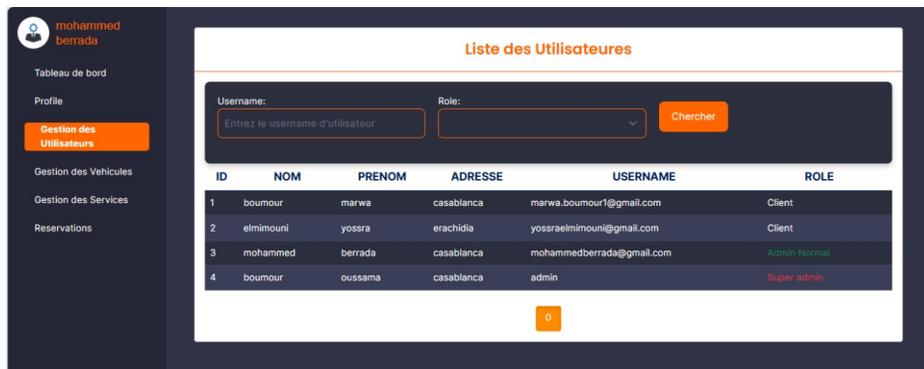
Adresse: casablanca

Username: mohammedberrada@gmail.com

Mot de passe: ...

Mettre à jour

Figure 40



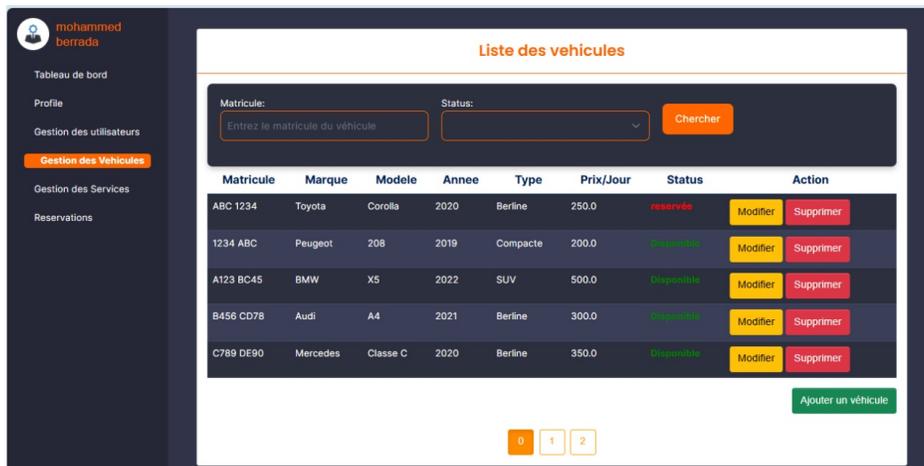
Liste des Utilisateurs

Username: Role: Chercher

ID	NOM	PRENOM	ADRESSE	USERNAME	ROLE
1	boumour	marwa	casablanca	marwa.boumourf@gmail.com	Client
2	elmimouni	yossra	erachidia	yossraelmimouni@gmail.com	Client
3	mohammed	berrada	casablanca	mohammedberrada@gmail.com	Admin Normal
4	boumour	oussama	casablanca	admin	Super admin

0

Figure 41



Liste des véhicules

Matricule: Status: Chercher

Matricule	Marque	Modele	Annee	Type	Prix/Jour	Status	Action
ABC 1234	Toyota	Corolla	2020	Berline	250.0	réservee	Modifier Supprimer
1234 ABC	Peugeot	208	2019	Compacte	200.0	disponible	Modifier Supprimer
A123 BC45	BMW	X5	2022	SUV	500.0	disponible	Modifier Supprimer
B456 CD78	Audi	A4	2021	Berline	300.0	disponible	Modifier Supprimer
C789 DE90	Mercedes	Classe C	2020	Berline	350.0	disponible	Modifier Supprimer

Ajouter un véhicule

0 1 2

Figure 42

Modifier un véhicule

Matricule

Marque

Modèle

Année

Type

Tarif Journalier

Statut



Image
 Aucun fichier choisi

Modifier

Figure 43

Liste des Services

Mot Clé: **Chercher**

ID	Nom	Prix/Jour	Image
1	Franchise Partielle	108,0	
2	Siège Bébé de 0 à 3 ans	70,0	
3	GPS	99,0	

Ajouter un Service

9

Figure 44

Ajouter un nouveau Service

Nom

Prix Journalier

Image
 Aucun fichier n'a été sélectionné

Ajouter

Annuler

Figure 45

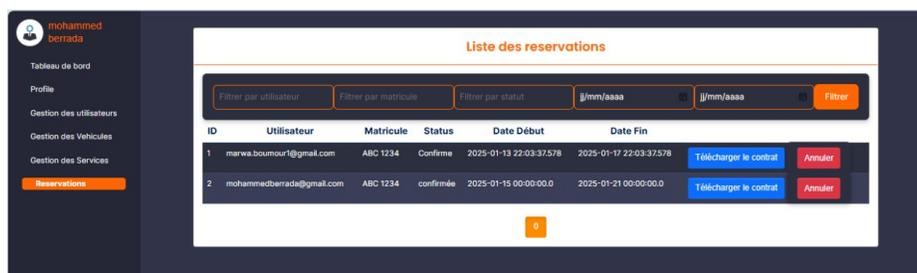


Figure 46

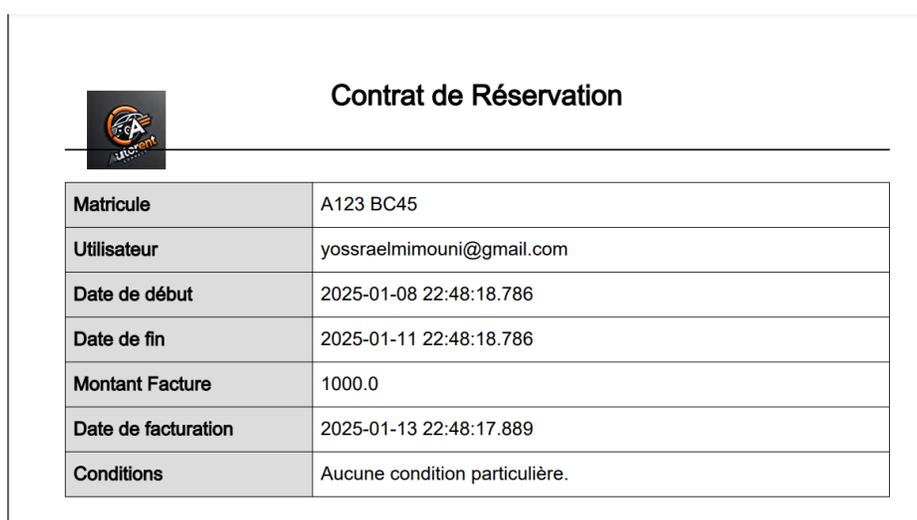


Figure 47: Après cliquer sur le bouton "Télécharger le contrat"

Conclusion

Dans ce rapport, nous avons présenté le processus de conception et de réalisation de notre application web, qui vise à simplifier et optimiser la gestion des réservations et des paiements dans le cadre de la location de voitures. En combinant des technologies modernes telles que Spring Boot, Spring Security, Thymeleaf pour le backend et des outils comme HTML, CSS, Bootstrap pour le frontend, nous avons pu créer une solution robuste et flexible, répondant aux besoins fonctionnels et techniques identifiés lors de l'analyse préalable.

L'application permet d'offrir une expérience utilisateur conviviale grâce à une interface intuitive et des fonctionnalités bien intégrées, telles que la gestion des utilisateurs, des véhicules, des réservations, ainsi que la génération automatique de documents contractuels. Nous avons également mis en place une architecture sécurisée grâce à Spring Security, garantissant une authentification fiable et une gestion des rôles adaptée (administrateurs et clients).

Ce projet nous a permis d'approfondir nos connaissances techniques et méthodologiques en développement logiciel, tout en nous confrontant aux défis liés à la conception d'une application

à vocation professionnelle. Nous avons également appris à collaborer efficacement au sein d'une équipe, à gérer les délais et à répondre aux exigences fonctionnelles dans un cadre structuré.

Pour aller plus loin, des améliorations futures pourraient inclure l'intégration de systèmes de paiement en ligne, l'ajout de fonctionnalités avancées telles que des recommandations personnalisées ou encore l'optimisation des performances pour une meilleure scalabilité.

En conclusion, ce projet représente une étape significative dans notre parcours en ingénierie informatique, et nous espérons que cette application pourra inspirer d'autres initiatives similaires dans le domaine de la location de services.

References

- [1] *Documentation de Spring Boot*, <https://spring.io/projects/spring-boot/>.
- [2] *Étude des principes de réservation de véhicules*, <https://www.hertz.ma/locations/vehicules>.
- [3] *Documentation de Bootstrap*, <https://getbootstrap.com/>.